Metaheuristic algorithms.

Lab 7: Neural networks with unsupervised learning. Clustering, topological mapping, combinatorial optimization.
Prediction using neural networks with contextual units (Elman networks)
Evolutionary training of neural networks

---

1. Neural networks with unsupervised learning

In the case of unsupervised learning, the training set contains only input data. The aim of the training algorithm is to identify correlations between these input dat and extract information concerning their structure (e.g. identify natural clusters in data).

The support architecture for unsupervised learning algorithms usually consists of a layer of input units and a layer of functional units. The most known neural networks models with unsupervised training are:

- Competitive networks (WTA and ART) – can be used for clustering tasks
- Kohonen networks – can be used for topological mapping (self-organized maps)

1.1. Clustering

Clustering means to identify natural groups in a set of data, such that the data in the same group are highly similar, while data in different groups are dissimilar. In the simplest clustering approaches, each group has a prototype (center) and the data are assigned to the cluster with the closest prototype. The process of identifying the prototypes can be done using a one-layer neural network having linear units and trained using a WTA (Winner Takes All) algorithm.

a) *Architecture.* To cluster a set of N-dimensional data in K clusters one can use a network with N input units and K output units. The matrix W containing the weights will have K rows and N columns. Each of the matrix rows can be interpreted as prototype of the cluster associated to the corresponding functional unit.
b) *Funcționing.* For each input data X is identified the winning unit k* satisfying $d(X,W^{k*})<d(X, W^{k})$ for any k=1..K. This means that X is assigned to cluster k*.
c) *Training.* For each example X from the training set, after the winning unit k* is identified its weights are adjusted according to: $W^{k*} = W^{k*}+ eta(X- W^{k*})$.

Application 1. Implement a WTA network for bidimensional data clustering.

Hint. See WTA.sci

1.2. Kohonen networks.

The main particularities of Kohonen networks with respect to the WTA networks are:

- The output units are placed in the nodes of a 1D, 2D or 3D grid. Each unit has associated a neighborhood. The neighborhood shape depends on the type of grid and on the distance used to define the neighborhood.
- For each element of the training set both the weights of the winning unit and the weights of the neighboring units are adjusted.

The Kohonen networks can be used to map high-dimensional data in smaller dimensional spaces such that the toplogical relations from the original space are preserved.

Application 2. Implement a Kohonen network with a rectangular grid and train it such that it maps data uniformly generated in [0,1]x[0,1].

Hint: See SOM.sci

Exercise. Retrain the network for data generated in the interior of a geometrical shape (e.g. circle, ring etc).

Elastic net = Kohonen network for solving the Travelling Salesman problem

Problem: given a set of nodes (locations, places, towns) find a tour which passes exactly once through each node and has a minimal cost (the cost is proportional to the length of the tour).

Main ideas:

- Network architecture:
  - input layer with 2 units
  - output layer with a ring topology (each neuron has a neighborhood of size 2s; the set of neighbours of unit i are: {i-s,i-(s-1),…,i-1,i,i+1,…,i+(s-1),i+s}.
- Thus each unit has two weights which are initialized randomly with values belonging to the space of values corresponding to the towns coordinates.
- Training: the list containing the coordinates of nodes is iterated for several times and at each iteration for each input vector x (pair of coordinates) there are several operations which are executed:
  - Find the winning unit, i*, (the unit which has the vector of weights closest to x)
  - Adjust the weights of all outout units using:
    - $W^i(k+1) = W^i(k) + \eta(X - W^i(k))\Lambda(i,i^*)$

      Where the neighborhood function is $\Lambda(i,i^*) = \exp(-(i-i^*)^2/(2s^2))$

Application 3. Solve a TSP problem using the network described above.

Hint. see Kohonen_TSP.sci

Exercise: Write a function which computes the tour length.

2. Prediction using neural networks with contextual units

The neural networks with contextual units are usually layered networks in which there exist several feedback connections (e.g. from the hidden units to some units from the input layer – this corresponds to the Elman network). The units receiving feedback signals are called contextual units and act as a short term memory.

In this way, when we want to design a network for predicting the next value in a scalar time series, it is enough to use one input unit, one output unit, a given number (e.g. K) of hidden units and the same number of contextual units. The contextual units receive copies of the signal produced by the hidden units, thus the weights of the feedback connections are fixed to the value 1. This is the architecture of an Elman network.

As only feedforward connections have trainable weights the training process can be based on the Backpropagation algorithm. (for details see Lecture 13, slides 32-33)

Application 4. Train an Elman network to predict the next value in a time series (e.g. the series corresponding to the currency exchange rate).

Hint. See predictionElman.sci

3. Evolutionary training of neural networks

Evolutionary algorithms (but also other population-based metaheuristics) can be used for neural networks design at least in two cases:

- Estimation of the synaptic weights in the case of networks with non-differentiable activation/transfer functions or in the case of recurrent networks (when traditional algorithms like Backpropagation cannot be applied)
- Establishing the architecture of the network (both in the case of explicit and in the case of implicit encoding – see Lecture 14) .

Application 5. Let us consider the problem of training a neural network in order to represent the XOR function.

We can use the following architecture:

- 2 input units + 1 dummy unit (used to provide the biases for the hidden units)
- K hidden units (K is an input parameter) with a Heaviside activation function + 1 dummy unit (used to provide the bias for the output unit)
- An output unit with a Heaviside activation function

Each element of the population will have 4*K+1 components corresponding to all weights and biases. The function to be minimized is the MSE (Mean Squared Error) computed for the training set containing all four examples: ((0,0),0), ((0,1),1),((1,0),1), ((1,1),0). Analyze the influence of the population sizes, mutation parameter, number of parents used for recombination, selection type (truncation or tournament) and value of K.

Hint. see SE_nn.m. The function used by the evolution strategy to evaluate a network is SE_XOR.

Homework. Implement an ART network (details in the slides of lecture 4) to identify clusters in a set of bidimensional data.
Hint. Start with WTA.sci and change the training algorithm